

How to use MFC to automate Excel 2000 and Excel 2002 and obtain an array from a range in Excel 2000 and Excel 2002

This article was previously published under Q186122

Article ID : 186122
Last Review : November 22, 2004
Revision : 5.0

SUMMARY

This article demonstrates how to automate Microsoft Excel and return the values from a multi-cell range to an array.

MORE INFORMATION

To return the values from a multi-cell range without querying the cells one-by-one, you must use the GetValue member function of the Range object to return the values to a VARIANT. You then construct a SAFEARRAY based on the VARIANT you returned from GetValue. Once you have the SAFEARRAY, you can access the elements using GetElement. The following steps illustrate this process and demonstrate how you can access both string and numeric elements.

Notes for Automating Microsoft Excel 2000 and 2002

The sample code in this article uses class wrappers generated from the Excel 97 object library (Excel 8.olb). With slight modification, this code can be applied to an Automation client that uses class wrappers for Excel 2000 (Excel9.olb) or Excel 2002 (Excel10.olb). For additional information about using the sample code described in this article with the Microsoft Excel 2000 or 2002 type library, click the article number below to view the article in the Microsoft Knowledge Base:

[224925](http://www.support.microsoft.com/kb/224925/EN-US/) (http://www.support.microsoft.com/kb/224925/EN-US/) INFO: Type Libraries for Office May Change with New Release

Steps to Create Project

1. In Microsoft Excel, create a new workbook. Populate cells A1:C8 with a mixture of string and numeric data. Save the workbook as "c:\test.xls" and exit Microsoft Excel.
2. Follow steps 1 through 12 in the following article in the Microsoft Knowledge Base to create a sample project that uses the IDispatch interfaces and member functions defined in the Excel8.olb type library:

[178749](http://www.support.microsoft.com/kb/178749/EN-US/) (http://www.support.microsoft.com/kb/178749/EN-US/)How To Create Automation Project Using MFC and a Type Library

3. At the top of the AutoProjectDlg.cpp file, add the following line:

```
#include "excel8.h"
```

4. Add the following code to CAutoProjectDlg::OnRun() in the AutoProjectDlg.cpp file:

Sample Code

```
// OLE Variant for Optional.
COleVariant VOptional((long)DISP_E_PARAMNOTFOUND, VT_ERROR);

_Application objApp;
_Workbook objBook;
_Workbooks objBooks;
_Worksheets objSheets;
_Worksheet objSheet;
_Range objRange;
VARIANT ret;

// Instantiate Excel and open an existing workbook.
objApp.CreateDispatch("Excel.Application");
objBooks = objApp.GetWorkbooks();
objBook = objBooks.Open("C:\\Test.xls",
    VOptional, VOptional, VOptional, VOptional,
    VOptional, VOptional, VOptional, VOptional,
    VOptional, VOptional, VOptional, VOptional);
```

```

objSheets = objBook.GetWorksheets();
objSheet = objSheets.GetItem(COLEVariant((short)1));

//Get the range object for which you wish to retrieve the
//data and then retrieve the data (as a variant array, ret).
objRange = objSheet.GetRange(COLEVariant("A1"), COLEVariant("C8"));
ret = objRange.GetValue();

//Create the SAFEARRAY from the VARIANT ret.
COleSafeArray sa(ret);

//Determine the array's dimensions.
long lNumRows;
long lNumCols;
sa.GetUBound(1, &lNumRows);
sa.GetUBound(2, &lNumCols);

//Display the elements in the SAFEARRAY.
long index[2];
VARIANT val;
int r, c;
TRACE("Contents of SafeArray\n");
TRACE("=====\n\t");
for(c=1;c<=lNumCols;c++)
{
    TRACE("\t\tCol %d", c);
}
TRACE("\n");
for(r=1;r<=lNumRows;r++)
{
    TRACE("Row %d", r);
    for(c=1;c<=lNumCols;c++)
    {
        index[0]=r;
        index[1]=c;
        sa.GetElement(index, &val);
        switch(val.vt)
        {
            case VT_R8:
            {
                TRACE("\t\t%1.2f", val.dblVal);
                break;
            }
            case VT_BSTR:
            {
                TRACE("\t\t%s", (CString)val.bstrVal);
                break;
            }
            case VT_EMPTY:
            {
                TRACE("\t\t<empty>");
                break;
            }
        }
    }
    TRACE("\n");
}

//Close the workbook without saving changes
//and quit Microsoft Excel.
objBook.Close(COLEVariant((short)FALSE), VOptional, VOptional);
objApp.Quit();

```

5. Compile the project and run it.
6. Click Run. The values in cells A1:C8 are displayed in the Output window.

REFERENCES

For additional information about automating Microsoft Excel using MFC, please click the article numbers below to view the articles in the Microsoft Knowledge Base:

[186120](http://www.support.microsoft.com/kb/186120/EN-US/) (http://www.support.microsoft.com/kb/186120/EN-US/) How To Use MFC to Automate Excel and Fill a Range with an Array

[184663](http://www.support.microsoft.com/kb/184663/EN-US/) (http://www.support.microsoft.com/kb/184663/EN-US/) How To Embed and Automate a Microsoft Excel Worksheet with MFC

[179706](http://www.support.microsoft.com/kb/179706/EN-US/) (http://www.support.microsoft.com/kb/179706/EN-US/) How To Use MFC to Automate Excel and Create/Format a New Workbook

[178781](http://www.support.microsoft.com/kb/178781/EN-US/) (http://www.support.microsoft.com/kb/178781/EN-US/) HOWTO: Automate Excel Using MFC and Worksheet Functions

[178783](http://www.support.microsoft.com/kb/178783/EN-US/) (http://www.support.microsoft.com/kb/178783/EN-US/) HOWTO: Use MFC to Create a Microsoft Excel Chart

[167668](http://www.support.microsoft.com/kb/167668/EN-US/) (http://www.support.microsoft.com/kb/167668/EN-US/) How To Pass a SafeArray of Strings in a VARIANT*

APPLIES TO

- Microsoft Excel 2000 Standard Edition
- Microsoft Visual C++ 5.0 Enterprise Edition
- Microsoft Visual C++ 6.0 Enterprise Edition
- Microsoft Visual C++ 5.0 Professional Edition
- Microsoft Visual C++ 6.0 Professional Edition
- Microsoft Visual C++ 6.0 Standard Edition
- Microsoft Excel 2002 Standard Edition
- Microsoft Excel 97 Standard Edition

Keywords: kbprogramming kbfunctions kbautomation kbhowto kbinterop KB186122